1.T0-DO-LIST:

```
No_of_tasks = []

def show_list():
   """Show the to-do list."""
   if not No_of_tasks:
      print("Your to-do list is an empty.")
   else:
      for i, tsk in enumerate(No_of_tasks, start=1):
         print(f"{i}. {tsk}")

def add(tsk):
   """Add a task to the to-do list."""
  No_of_ tasks.append(tsk)
   print(f"Added task: {tsk}")

def remove(tsk_number):
   """Remove a task from the to-do list."""
   if 1 <= tsk_number <= len(No_of_tasks):
      removed_tsk =No_of_ tasks.pop(tsk_number - 1)
      print(f"Removed tsk: {removed_tsk}")
   else:
      print("Invalid task number. Please check your to-do list.")

while True:
   print("\nChoose an option:")
   print("1. Show to-do list")
   print("2. Add a task")
   print("3. Remove a task")
   print("4. Quit")

   choice = input("Enter the number of your choice: ")

   if choice == '1':
      show_list()
   elif choice == '2':
      tsk = input("Enter the task: ")
      add(tsk)
   elif choice == '3':
      tsk_number = int(input("Enter the task num to remove: "))
      remove(task_number)
   elif choice == '4':
      print("Exiting the to-do list application. Goodbye!")
      break
   else:
      print("Invalid choice. Please enter a valid number.")
```

2.**CALCULATOR:**

```
def simple_calculator():
   try:
      number1 = float(input("Enter the first number: "))
      operator = input("Enter the arithmetic operation (+, -, *, /): ")
      number2 = float(input("Enter the second number: "))
```

```
        if operator == "+":
            res_of_opr = number1 + number2
        elif operator == "-":
            res_of_opr = number1 - number2
        elif operator == "*":
            res_of_opr = number1 * number2
        elif operator == "/":
            if number2 != 0:
                res_of_opr = number1 / number2
            else:
                print("Error: Division by zero is not allowed.")
                return
        else:
            print("Invalid operator. Please enter  valid one +, -, *, or /.")
            return

        print(f"Result: {res_of_opr}")

    except ValueError:
        print("Invalid input. Please enter valid numbers.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Run the calculator
simple_calculator()
```

3. **PASSWORD:**

```
import random
import string

def generate_the_password(len):
    random_characters = string.ascii_letters + string.digits + string.punctuation
    password_to_generate = ''.join(random.choice(random_characters) for _ in range(len))
    return password_to_generate

def pswd_generator():
    try:
        len = int(input("Enter the desired length of the password: "))
        if len <= 0:
            print("Please enter a positive length.")
            return

        password_to_generate = generate_the_password(len)
        print(f"Generated Password: {password_to_generate}")

    except ValueError:
        print("Invalid input. Please enter a valid positive integer for the password length.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Run the password generator
pswd_generator()
```

4. ROCK-PAPER-SEASOR:

```
import random

def user_choice():
  while True:
    get_user_choice = input("Choose rock, paper, or scissors: ").lower()
    if get_user_choice in ['rock', 'paper', 'scissors']:
      return get_user_choice
    else:
      print("Invalid choice. Please choose rock, paper, or scissors.")

def determine_the_winner(get_user_choice, computer_choice):
  if get_user_choice == computer_choice:
    return "This game is  tie!"
  elif (
    (get_user_choice == 'rock' and computer_choice == 'scissors') or
    (get_user_choice == 'scissors' and computer_choice == 'paper') or
    (get_user_choice == 'paper' and computer_choice == 'rock')
  ):
    return "You win the game!"
  else:
    return "You lose the game!"

def rock_paper_scissors_game():
  user's_score = 0
  computer's_score = 0

  while True:
    print("\n Rock, Paper, Scissors Game")
    print("---------------------------")

    get_user_choice = user_choice()
    computer_choice = random.choice(['rock', 'paper', 'scissors'])

    print(f"\n Your choice is: {get_user_choice}")
    print(f"Computer's choice is: {computer_choice}")

    result = determine_the_winner(get_user_choice, computer_choice)
    print(f"\nResult of the game is: {result}")

    if 'win' in result:
      user's_score += 1
    elif 'lose' in result:
      computer's_score += 1

    print(f"\n Score - You: {user's_score} | Computer: {computer's_score}")

    play_the_game_again = input("\nDo you want to play game again? (yes/no): ").lower()
    if play_the_game_again != 'yes':
      print("Thanks for playing the game! bye.")
      break

# Run the rock-paper-scissors game
rock_paper_scissors_game()
```

5. **CONTACT:**

```
 class ContactInformation:
  def _init_(self, name, ph_num, email, address):
    self.name = name
    self.ph_num = phone_number
    self.email = emailaddress
    self.address = address

class Add_ContactManager:
  def _init_(self):
    self.contacts = []

  def add_the_contact(self, contact):
    self.contacts.append(contact)
    print(f"Contact {contact.name} added successfully!")

  def view_contact_list(self):
    print("\n the Contact List:")
    for contact in self.contacts:
      print(f"Name: {contact.name}, Phone: {contact.phone_number}")

  def search_the_contact(self, search_term):
    results = [contact for contact in self.contacts if
          search_term.lower() in contact.name.lower() or search_term in contact.ph_num]
    return results

  def update_the_contact(self, contact_name):
    for contact in self.contacts:
      if contact.name.lower() == contact_name.lower():
        new_phone_number = input(f"Enter new phone number for {contact_name}: ")
        contact.ph_num = new_phone_number
        print(f"Contact {contact_name} updated successfully!")

  def delete_the_contact(self, contact_name):
    self.contacts = [contact for contact in self.contacts if contact.name.lower() !=
contact_name.lower()]
    print(f"Contact {contact_name} deleted successfully!")


def main():
  contact_manager = Add_ContactManager()

  while True:
    print("\n Contact Management System")
    print("1. Add Contact")
    print("2. View Contact List")
    print("3. Search Contact")
    print("4. Update Contact")
    print("5. Delete Contact")
    print("6. Exit")

    choice = input("Enter your choice from(1-6): ")

    if choice == '1':
      name = input("Enter contact name: ")
      ph_num = input("Enter phone number: ")
      email = input("Enter emailaddress: ")
```

```
            address = input("Enter  the address: ")

            new_contact = Contact(name, ph_num, email, address)
            contact_manager.add_the_contact(new_contact)

        elif choice == '2':
            contact_manager.view_contact_list()

        elif choice == '3':
            search_term = input("Enter name or phone number to search: ")
            search_results = contact_manager.search_the_contact(search_term)
            if search_results:
                print("\n Search Results:")
                for contact in search_results:
                    print(f"Name: {contact.name}, Phone: {contact.ph_num}")
            else:
                print("No matching contacts found.")

        elif choice == '4':
            contact_name = input("Enter the name of the contact to update: ")
            contact_manager.update_the_contact(contact_name)

        elif choice == '5':
            contact_name = input("Enter the name of the contact to delete: ")
            contact_manager.delete_the_contact(contact_name)

        elif choice == '6':
            print("Exiting Contact Management System. Goodbye!")
            break

        else:
            print("Invalid choice. Please enter a number from 1 to 6.")

if _name_ == "_main_":
    main()
```